Part 2.3: Visualization

Jakob Weiss

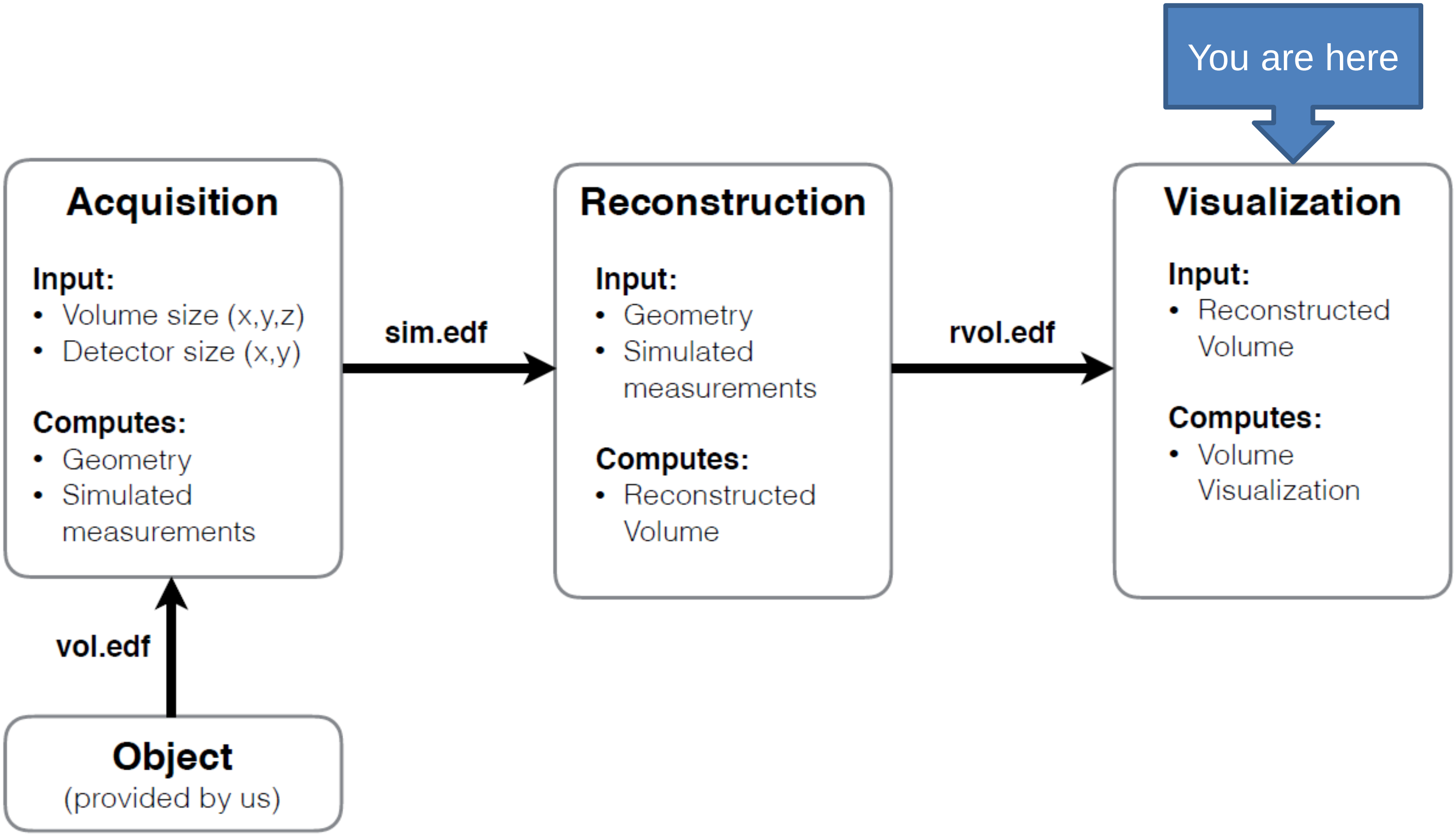
Slides based on material by Christian Schulte zu Berge

Technische Universität München

Johns Hopkins Whiting School of Engineering

OVERVIEW

Part 2 - Overview

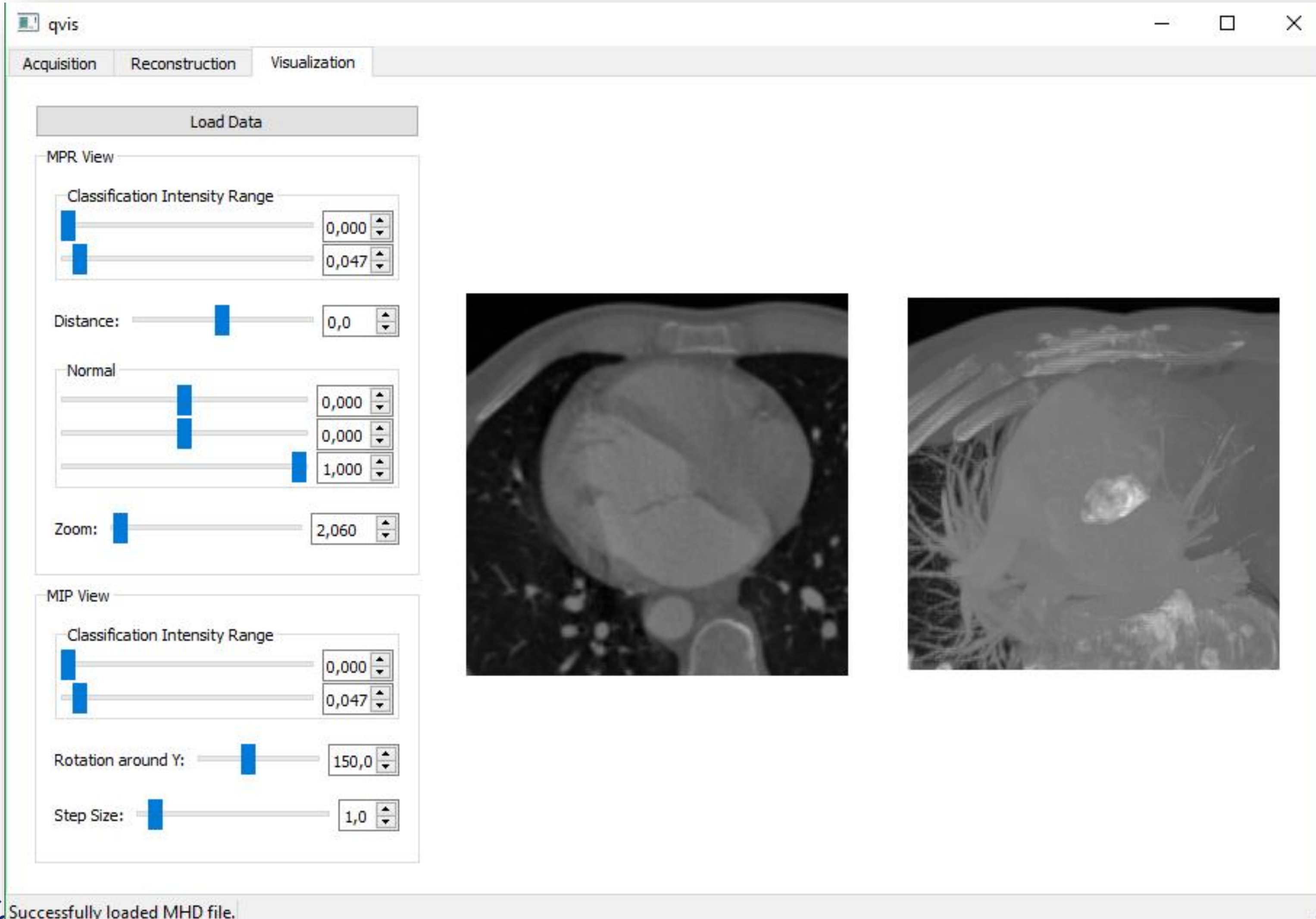


Visualization Part

You will:

- Gain some background knowledge on Medical Visualization
- Learn some essential techniques:
  - Trilinear interpolation
  - Classification (transfer functions)
  - Slice visualization
  - Simple raycasting-based volume visualization
- Create your own volume visualization tool

Example Application



Disclaimer

- Today, most visualization techniques are implemented entirely on the GPU (e.g. OpenGL, DirectX, …)
- However:
  - This is a C++ course, not a Computer Graphics course.
  - Everything will be implemented on the CPU.
  - (☹)

Introduction Into Visualization

Technische Universität München

Johns Hopkins Whiting School of Engineering

Definition

vi·su·al·ize[1]

- To form a mental image of; envisage
- To make visible

McCormick et al.[2]

- „Visualization is a method of computing. It transforms the symbolic into the geometric, enabling researchers to observe their simulations and computations. Visualization offers a method for seeing the unseen."

1. The American Heritage Dictionary of the English Language
2. Visualization in Scientific Computing, Computer Graphics 21(6), Nov 1987

Disambiguation

Computer Graphics ↔ Visualization

- Both transform data into graphics/pictures
- CG: Focus on rendering, shading, texturing
- Vis: Focus on data abstraction and representation

Further Differentiations

- Efficient algorithms (CG) ↔ Effective to use (Vis)
- Mapping image to abstract information (CV) ↔ Mapping information to image (Vis)
- Mapping data to data (IP) ↔ Mapping data to image (Vis)
- Perception Science ↔ Use this knowledge for better visualization (Vis)

Data Characteristics

Source

- Real world (measuring) ↔ Theory (simulation) ↔ Art (design, modelling)

Domain

- Nominal ↔ Ordinal ↔ Discrete metric ↔ Continuous metric

Dimensionality

- Scalar ↔ Vector ↔ Tensor ↔ Multivariate

Structure

- Regular grid ↔ Scattered data

Data Characteristics in our Application

Source

- Real world (measuring) ↔ Theory (simulation) ↔ Art (design, modelling)

Domain

- Nominal ↔ Ordinal ↔ Discrete metric ↔ Continuous metric

Dimensionality

- Scalar ↔ Vector ↔ Tensor ↔ Multivariate

Structure

- Regular grid ↔ Scattered data

Visualization Techniques for Scalar Data

- Dimensionality matters:
  - 1D scalar field: $\Omega \in \mathbb{R} \rightarrow \mathbb{R}$
  - 2D scalar field: $\Omega \in \mathbb{R}^2 \rightarrow \mathbb{R}$
  - 3D scalar field: $\Omega \in \mathbb{R}^3 \rightarrow \mathbb{R}$

- Additional time domain?
  - E.g. 2D+t scalar field: $\Omega \in \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}$

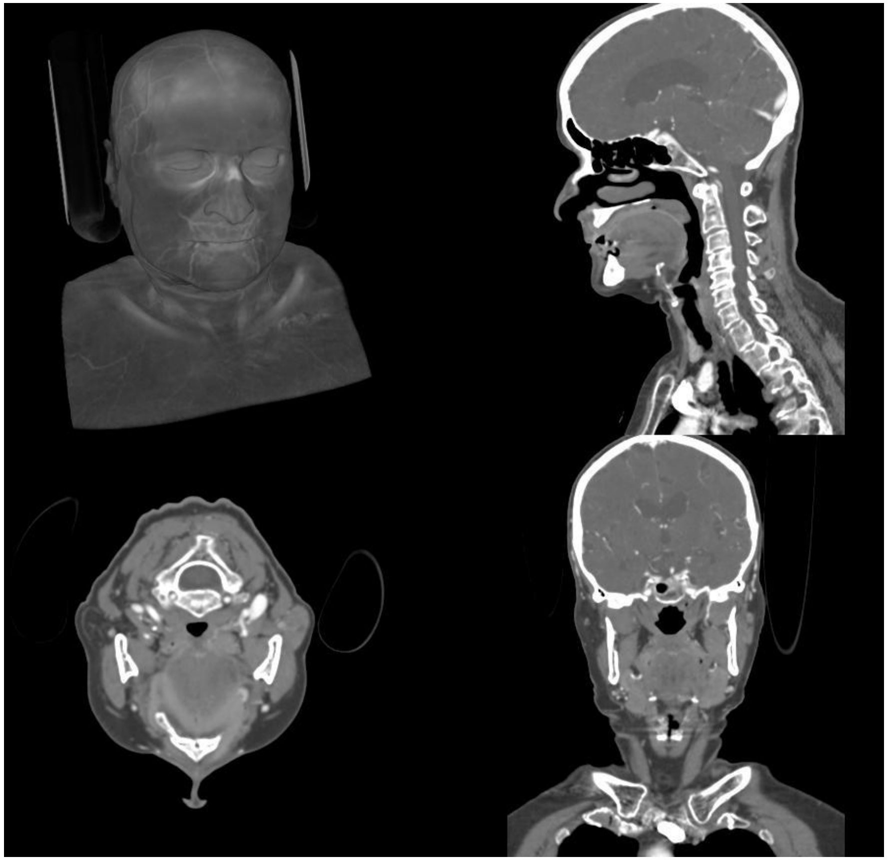
- Basic strategies:
  - Mapping to geometry (Function plots, height fields, …)
  - Mapping to color (Slice rendering, volume rendering)

$\rightarrow$ Visualization method depends heavily on data and application

Slice and Volume Visualization

Most medical data (CT, MRI, US, …) has:

- Discrete metric data
- Scalar values
- Regular 2D/3D grid



Standard visualization techniques:

- 2D: Slice rendering
  - Axis aligned slice
  - Multi-planar reconstruction (MPR)
    - *Natively supported by today's GPUs*
- 3D: Volume rendering
  - Indirect Volume Rendering
    - *First, extract geometry, then, render geometry*
  - Direct Volume Rendering
    - *Directly project volume onto 2D screen by simulating physics of light transport*

Essential Technique #1:
Trilinear Interpolation

Technische Universität München
Johns Hopkins
Whiting School of Engineering

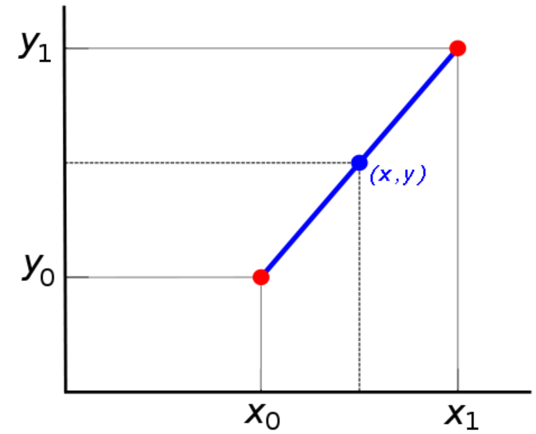
Interpolation

- Why do we need interpolation?
  - Image data has discrete representation: Voxel grid
  - Ray samples may lie between the voxels
- Nearest neighbor
  - Poor quality
- (Bi-/Tri-)Linear interpolation
  - Acceptable quality, fast implementation

Linear Interpolation

- Given two points $(x_0, y_0), (x_1, y_1)$, the following equation holds for any (x, y) on the straight line in between:

$$\frac{y - y_0}{x - x_0} = \frac{y_1 - y_0}{x_1 - x_0}$$



- Solving for y yields:

$$y = y_0 + (y_1 - y_0)\frac{x - x_0}{x_1 - x_0}$$

- We can generalize for any 1D function f:

$$f(x) \approx (1-t)f(x_0) + tf(x_1), \qquad t = \frac{x - x_1}{x_2 - x_1} \in [0, 1]$$

Image courtesy of: wikipedia.org

Bilinear Interpolation

Extending this to 2D functions yields bilinear interpolation:

- First, perform two linear interpolations in x direction

$$f(R_1) \approx (1 - t_x) f(Q_{11}) + t_x f(Q_{21})$$
$$f(R_2) \approx (1 - t_x) f(Q_{12}) + t_x f(Q_{22})$$

- Then, perform interpolation in y direction

$$f(P) \approx \left(1 - t_y\right) f(R_1) + t_y f(R_2)$$

→ Three linear interpolations in total

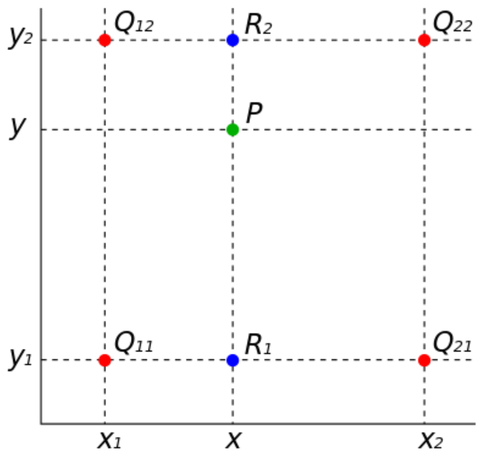


Image courtesy of: wikipedia.org

Trilinear Interpolation

Extension to 3D functions yields trilinear interpolation:

- First, perform two bilinear interpolations in the xy plane to yield $f(C_0)$ and $f(C_1)$

- Then, perform interpolation in z direction

$$f(C) \approx (1 - t_z) f(C_0) + t_z f(C_1)$$

$\rightarrow$ Seven linear interpolations in total

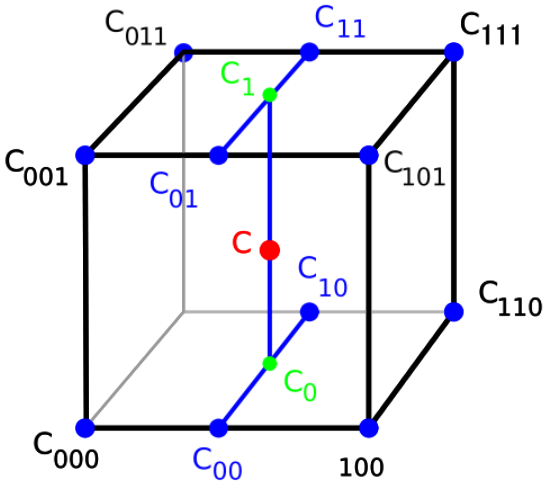


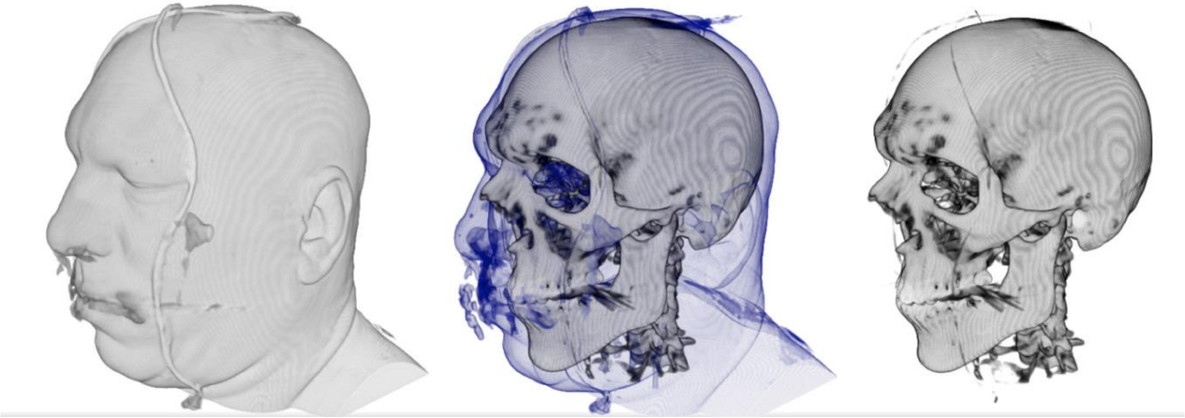
Image courtesy of: wikipedia.org

ESSENTIAL TECHNIQUE #2:
CLASSIFICATION / TRANSFER FUNCTIONS

Slice and Volume Visualization: Classification

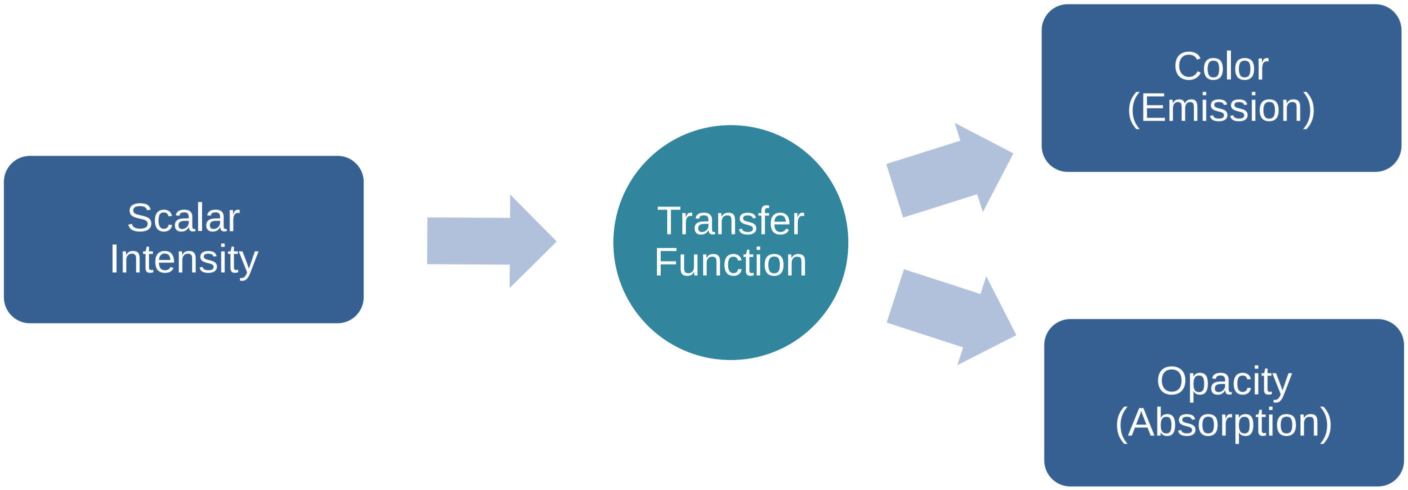
- Medical images usually only have scalar intensities as domain

- How can we define the look of the data?

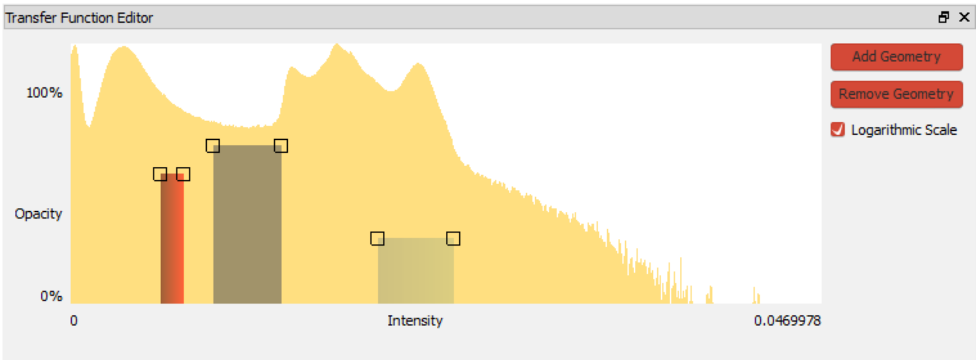


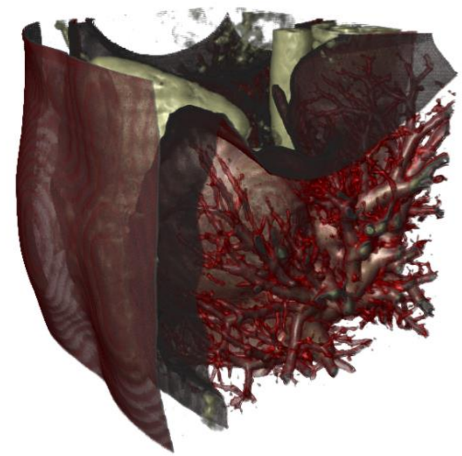
Classification: Transfer Functions

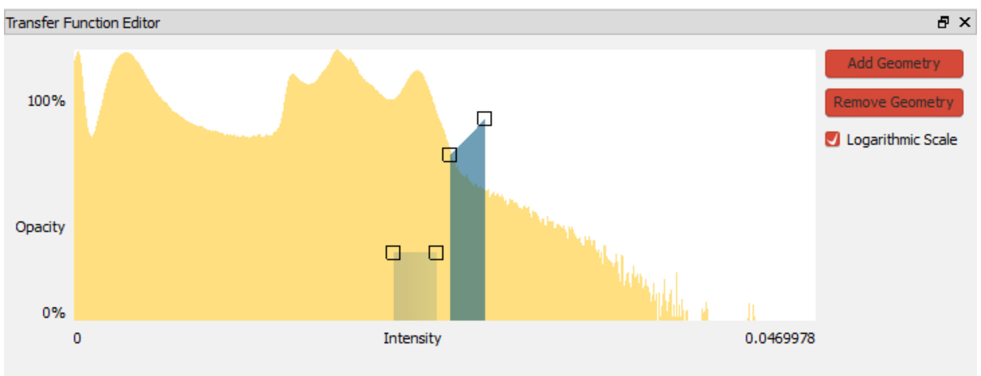
- Simple concept:
  Transfer functions map scalar intensities to optical properties

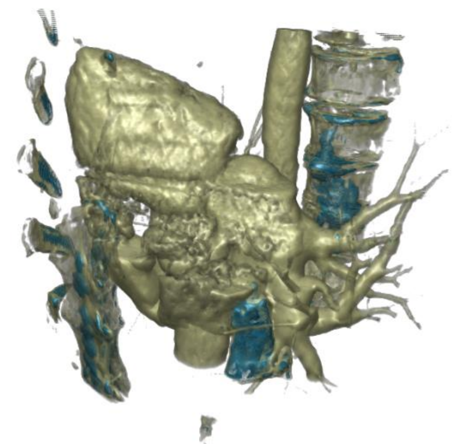


Classification: Transfer Function Examples









Essential Technique #3:
Multi-planar Reformation (MPR)

Technische Universität München

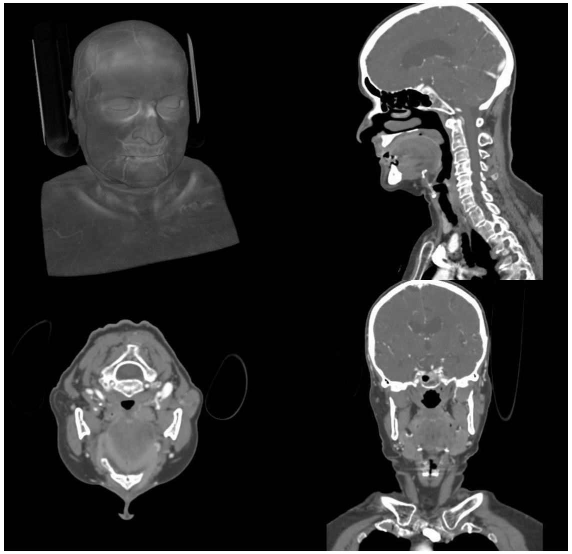
Johns Hopkins Whiting School of Engineering

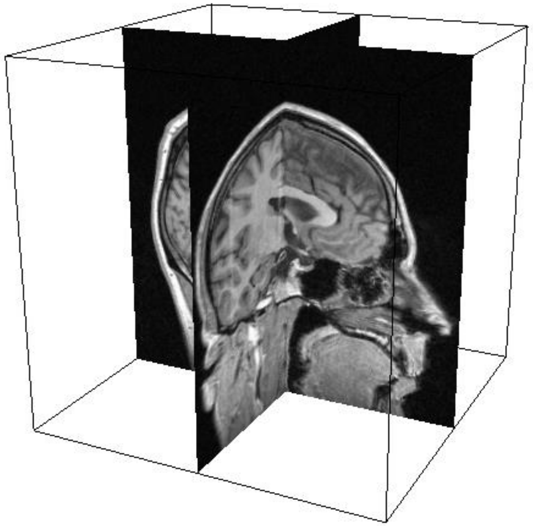
2D Visualization of 3D Volumes

- A 2D view shows only a subset of the 3D data
- Show a cut plane through the volume

- Simplest case:
  - Axis-aligned slice in original resolution
  - → can use voxel intensities directly

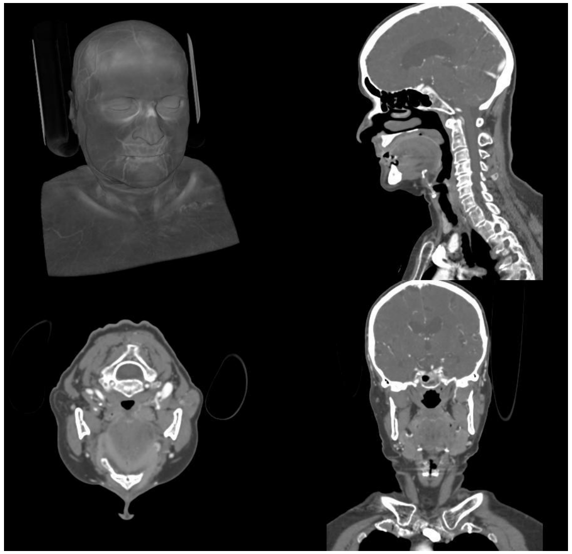
- Generic case:
  - Arbitrary slice orientation
  - Slice resolution != volume resolution
  - → need interpolation

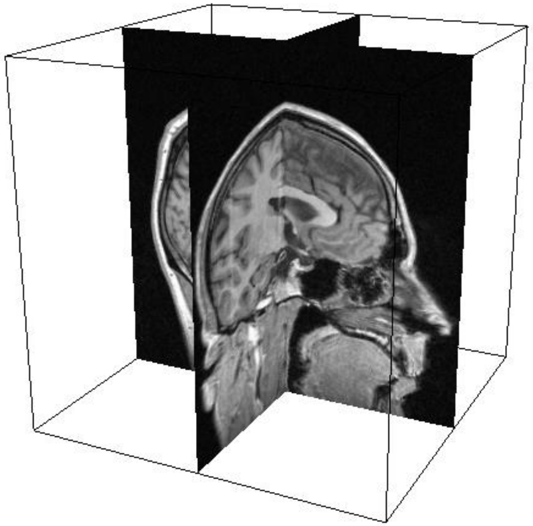




2D Visualization of 3D Volumes

- Multi-planar Reformation (MPR)
  - Define a cut plane through volume (e.g. use Hesse Normal Form)
  - Find transformation from 2D pixel space to voxel space
  - For each pixel:
    - Compute corresponding voxel
    - Acquire voxel intensity (interpolation!)
    - Apply transfer function
    - Color pixel accordingly

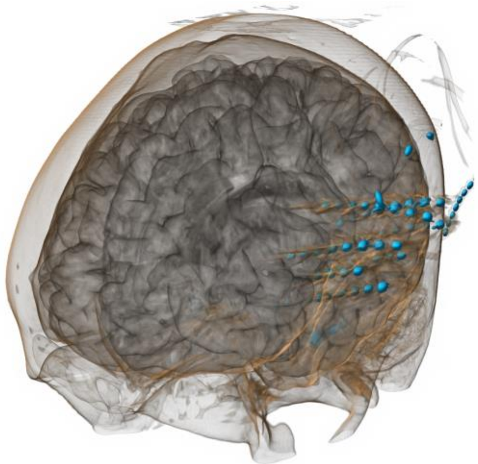


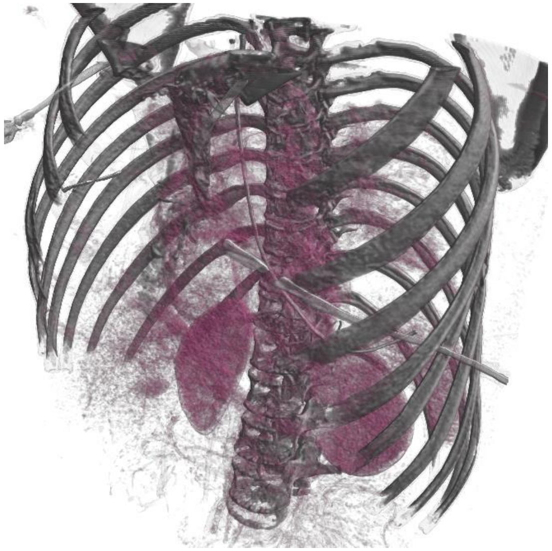


Essential Technique #4:
Direct Volume Rendering (Ray Casting)

Direct Volume Rendering

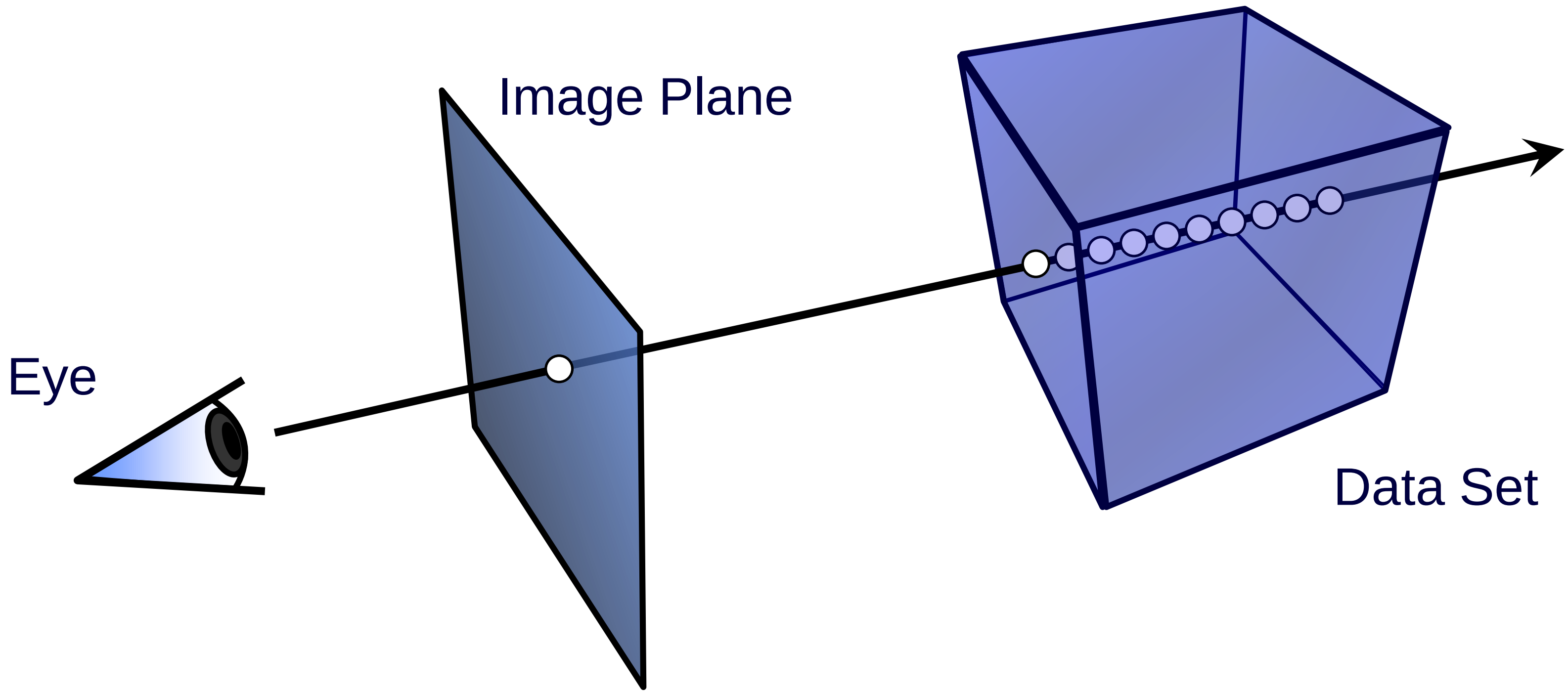
- How do we get a photo-realistic rendering of the data?





Sampling → Classification → Compositing

The Direct Volume Rendering Pipeline

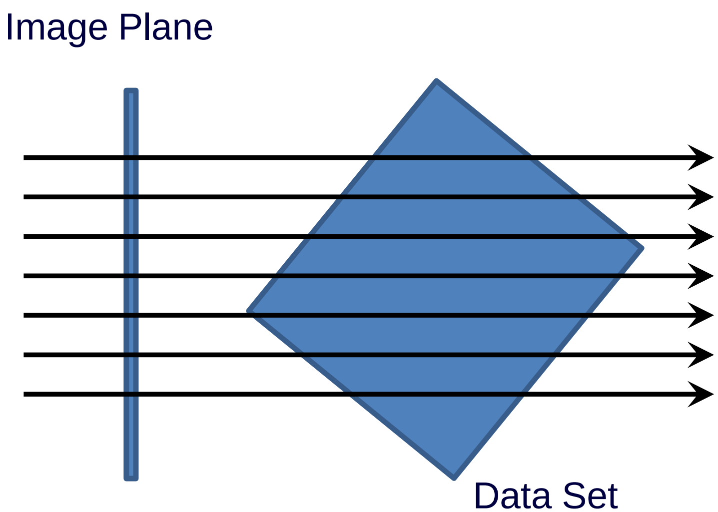


- Direct Volume Rendering
  - Setup a scene with a virtual camera and image plane
  - Simulate physics of light transport through ray casting
  - Sample voxel intensities along rays (interpolation)
  - Integrate voxel intensities along rays (compositing)

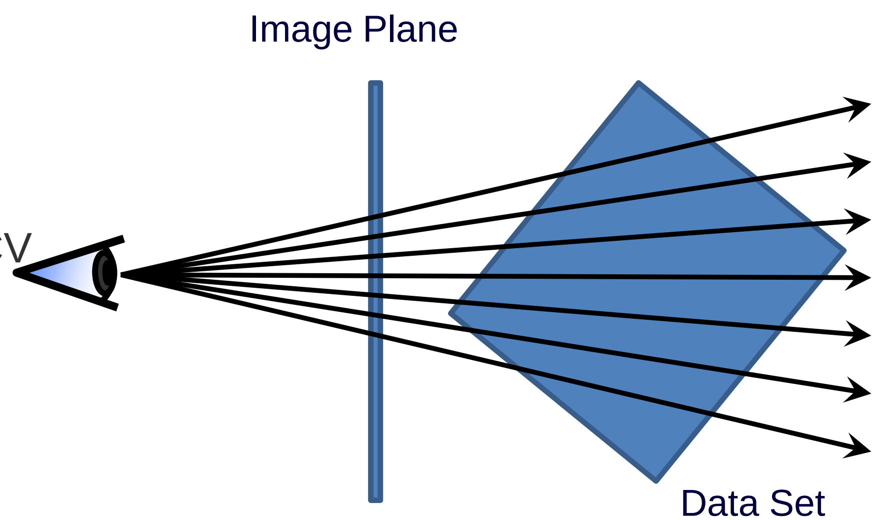
Image courtesy of: Engel et al.: „Real-Time Volume Graphics Tutorial“ – *Eurographics Workshop 2006*

Orthographic vs. Perspective Projection

- Orthographic Projection
  - Often used in engineering drawings
  - Not a natural camera model



- Perspective Projection
  - Fits the eye’s natural imaging process better by modeling a pinhole camera
  - More correct, usually used in CG, Vis, CV



Compositing Modes

- The compositing mode (in conjunction with the transfer function) defines the look of the data

- Maximum Intensity Projection (MIP)
  - Simplest form, just use the maximum intensity along each ray
- Direct Volume Rendering (DVR)
  - Photo-realistic rendering, approximates physics of light transport
- Digitally Reconstructed Radiograph (DRR)
  - Simulate 2D X-Ray images from CT
- …

ASSIGNMENT 9

Your Assignment

- Create a tool to visualize volume data
- → Use this to inspect and gain insight into the mystery data

- Functionality to implement:
  - Trilinear interpolation (~10 LOC)
  - Simple classification scheme (<10 LOC)
  - 2D MPR rendering (~30 LOC)
  - Simple Direct Volume Rendering with MIP classification scheme (~50 LOC)

Optional Tasks

- Use advanced Transfer functions
  - You can use a transfer function editor from the internets if you like, as long as you give credit

- Use advanced Compositing schemes
  - DRR is simple to adapt from the simulation in part 1
  - DVR requires a front-to-back alpha blending
  - DVR without illumination might still look crappy
    - Use the volume gradient as a „surface normal“
    - Lambertian diffuse shading or Blinn, Phong, or Blinn-Phong shading

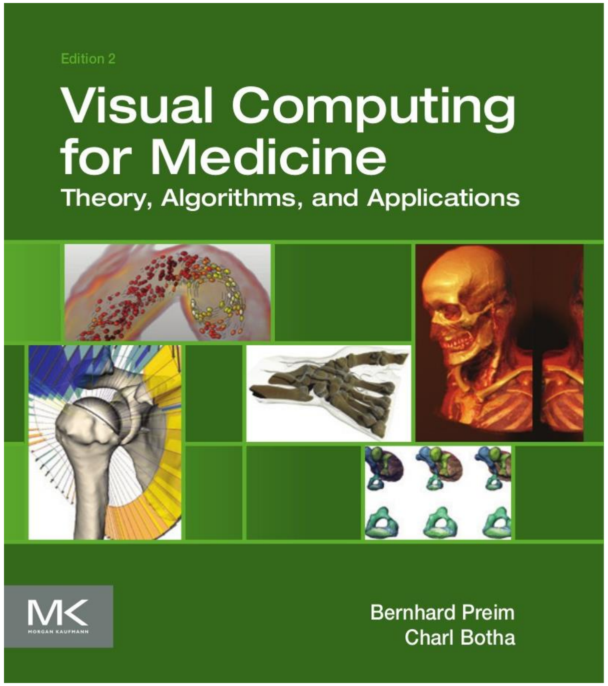
- Use parallelization
  - CPU-based Volume Rendering is slow
  - It scales with the display resolution
  - OpenMP, tbb and std::thread are nice ways to achieve parallelization

FURTHER READING

Further Reading

Survey Book (goes far beyond this seminar):

- Excellent overview over almost the entire body of current literature on medical visualization
  → http://medvisbook.com



More in-depth information on the different topics:

- English version of Prof. Preim's (University of Magdeburg) lecture slides on Medical Visualization are available at:
  → http://medvisbook.com/courses-and-tutorials/prof-preims-international-medvis-lectures/

Important!

Deadline is coming!

February 1st 2017

- Report to Tobias about your progress
- Make sure everything compiles on the VM/CI
- Make sure everybody did something

Thank You

Technische Universität München

Johns Hopkins
Whiting School
of Engineering